Government Software Projects Rank High in Major Critical Success Factors

Capers Jones
Software Productivity Research Inc., Artemis Management Systems

Capers Jones, one of the judges for CROSSTALK*'s first "Top 5 Quality Software Projects" awards, compares the projects he reviewed for the contest with those he has recently seen embroiled in legal disputes in the "real world." He comes away refreshed to see proof in these government projects that large and complex software projects can be finished on time, within budget, meet with favorable user reactions, and have few remaining defects after delivery.*

Having just returned from working as an expert witness in a software breach of contract lawsuit, it was very refreshing to read all 16 of the finalists' project descriptions submitted for the evaluation for CROSSTALK's Top 5 Quality Software Projects awards. None of the 16 projects made the kinds of mistakes in project management and quality control that have kept me in various courtrooms during the past few years.

Since most of the software projects that end up in court exceed their cost estimates by several hundred percent, have distressingly poor quality control, and bungle project management tasks such as sizing and schedule planning, it was quite enjoyable to read how competent software vendors go about building successful packages. All 16 teams are to be congratulated. In particular, teams with projects larger than 10,000 function points or 1 million single lines of code (SLOC) deserve a great deal of credit.

Software projects are influenced by more than 100 different factors. However, when similar projects are examined where one is successful (i.e., on time with good quality) and one is a failure (i.e., cancelled, delayed, or inoperable) about a dozen key factors tend to distinguish success from failure. Since there are many more large-project failures than successes, these were a rare breed.

The accompanying sidebar lists major factors associated with both success and failure that I have noted in examining many thousands of software projects. The list is taken from my book *Patterns of Software System Failure and Success*, International Thomson Computer Press 1995.

Of the 16 finalist projects submitted, all were better than average in every one of these critical factors. The top five projects ranged from "very good" to "outstanding" in all of these critical factors.

Another observation from reviewing the results of the nominations is the solid evidence that ascending the Software Engineering Institute's Capability Maturity Model® up to Level 3 or higher is well worthwhile. Indeed, for really large applications in the range of 10,000 function points or 1million SLOC, Level 5 is the desirable level for optimal performance.

There are so many software overruns and outright cancellations that it was quite refreshing to see "existence proofs" that large and complex software projects can be finished on time, within budget, meet with favorable user reactions, and have few remaining defects after delivery.

This evaluation of top software projects is so useful that I think CROSSTALK should be commended. My personal hope is that similar evaluations of top software projects will continue to be carried out annually.

It is hard to learn much from average projects. We can learn things from failures and disasters, of course, but we do not want to pattern our own projects on unfortunate models. The project descriptions and results submitted for this award provide one of the best models for building future projects that I have seen in many years. There are hundreds of ways to botch up projects, and only a few ways to build them successfully. Excellence in project management, estimating, measurement, quality control, and change control are all required for successful results. All of the projects submitted are to be commended, and the top five deserve accolades from the software community.◆

Successful Projects

- Effective project planning.
- Effective project cost estimating.
- Effective project measurements.
- Effective project milestone tracking.
- Effective project quality control.
- Effective project change management.
- Effective development processes.
- Effective communications.
- Capable project managers.
- Capable technical personnel.
- Significant use of specialists.
- Substantial volume of reusable materials.

Failing Projects

- Inadequate project planning.
- Inadequate project cost estimating.
- Inadequate project measurements.
- Inadequate project milestone tracking.
- Inadequate project quality control.
- Ineffective project change management.
- Ineffective development processes.
- Ineffective communications.
- Ineffective project managers.
- Inexperienced technical personnel.
- Generalists rather than specialists.
- Little or no reuse of technical material.